Geosensing Engineering and Mapping (GEM)
Civil and Coastal Engineering Department
University of Florida

Tutorial: Basic Gridding of ALSM Point Data

GEM Center Report No. Rep_2005-01-003

M. Starek and K.C. Slatton

January 12, 2005

Point of Contact:
Prof. K. C. Slatton
University of Florida; PO Box 116130; Gainesville, FL 32611
Tel: 352.392.0634, Fax: 352.392.0044, E-mail: slatton@ece.ufl.edu

Tutorial: Basic Gridding of ALSM Point Data

1.0 Introduction

This tutorial presents a MatLab function called "GridData1.m" that allows the user to grid Airborne Laser Swath Mapping (ALSM) point data. Either 1st-return or last-return intensities or heights may be gridded. The input is assumed to be the nominal 9-column ASCII data file produced during the ALSM processing by the Realm software package.

The user can specify the interpolation method and grid size in the function call. The program prompts the user for which data they want gridded. The program returns the x,y,z values and a 3-D surface plot. The function header explains how to call the function and run it. The user can access this help info by typing "help GridData1" in the MatLab command prompt.

The code uses MatLab's built in "griddata" function to grid the data by interpolating to a regular grid utilizing one of MatLab's built in interpolation methods specified by the user. Appendix A presents an example of 9-column data and an explanation of the data format. Appendix B displays the code for the "GridData1" function.

2.0 Basics of Gridding

Gridding is the process of converting an irregularly distributed point data set into a regular, equally spaced, rectangular grid of points commonly referred to as a raster array. In the case of ALSM data, the measures being gridded are usually the elevations (z-values) as is done when creating a digital elevation model (DEM), but other values such as the intensity measures of the returns can be gridded as well. The value of each grid cell is located at the center of the cell as observed in Figure 1 below. The nominal cell spacing between adjacent grid cells is referred to as the spacing or resolution of the grid and is often set to 1m for high-resolution ALSM data.

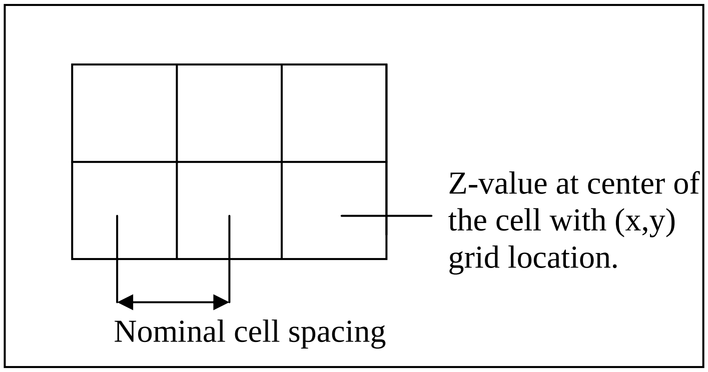

Figure 1: Example of a Regular Grid

To convert from irregular data to a regular grid, the data must be resampled by interpolation. Numerous interpolation methods exist and each method has positive and negative attributes dependent upon the data and desired results. Methods include linear, nearest neighbor, and Kriging. For the "GridData1" function, the user can select from one of MatLab's built in interpolation functions: linear, inverse-distance, nearest neighbor, or cubic spline.

For example, in figure 2 below, the points labeled 1, 2, and 3 are the actual, irregular data points whose (x,y) values, easting and northing for 9-column ALSM data, fall within that grid cell. Those data points and their z-value are then used to interpolate a value for that grid cell at the center of the cell. If nearest neighbor interpolation was selected, then the grid cell would be assigned the z-value associated to point 2 since that point is closest to the (x,y) location of the grid cell center as observed in the figure. Although only the points within the cell were used to interpolate for this example, it is important to mention that methods such as Kriging, cubic spline, and others, can use data points both within and outside the range of the cell.

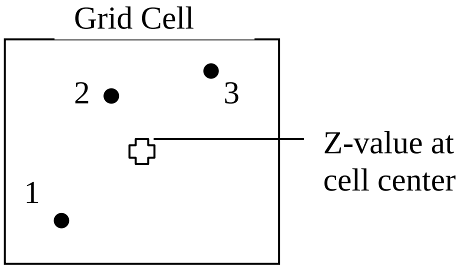

Figure 2: Example of Cell Interpolation

Advantages of the data being represented as a regular grid include ease of visualization, relatively simple computer processing, and enabling of frequency analysis. Disadvantages include the possibility of missing critical points in the data set and flat areas with little change can cause waste of storage if represented as a regular grid. Figure 3 displays an example of a 3D plot generated by the "GridData1" function for a linearly interpolated, gridded ALSM data set.

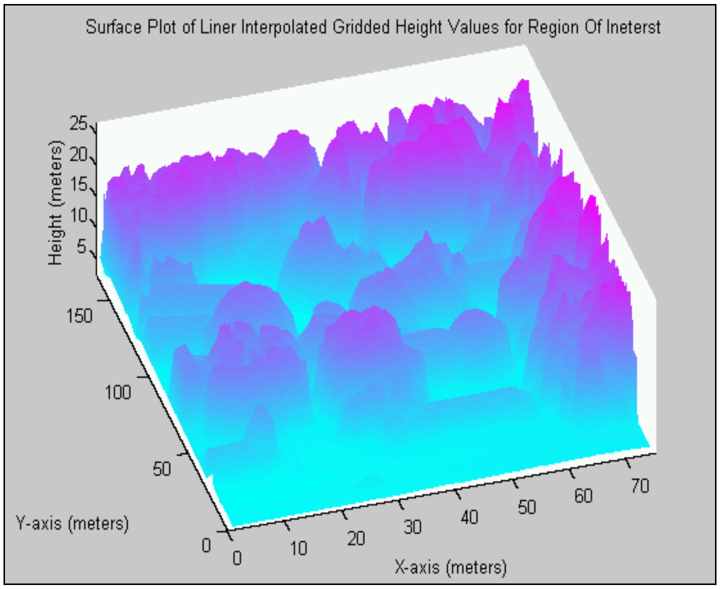

Figure 3: Example of a Last Stop surface Plot

Appendix A

Example of 9-column ALSM:

```
   C1              C2         C3        C4    C5        C6        C7        C8      C9
491349.921719 17367085.15 3281907.81   13.59   100 17367085.15 3281907.82   13.62   100
491349.921749 17367086.51 3281907.89   14.79    70
491349.921779 17367087.49 3281907.94   15.55   109 17367087.49 3281907.94   15.53   109
491349.921809 17367086.93 3281907.69    9.24     9 17367089.01 3281908.03   16.83    91
491349.921839 17367085.20 3281907.31    0.04     7 17367089.85 3281908.06   17.15   101
```

- Column 1 (C1): time stamp (seconds)

- Column 2 (C2): 2nd return x-coordinate which is in Universal Transverse Mercator (UTM) projection easting values in meters; the number 17 at the end of each value is the UTM zone code and must be removed by subtracting 17000000 from each data value to give the actual easting value. The zone code will change dependent upon the geographic locale of the data set.

- Column 3 (C3): 2nd return y-coordinate; UTM northing in meters.

- Column 4 (C4): 2nd return z-value; elevation in meters

- Column 5 (C5): 2nd return intensity

- Column 6 (C6): 1st return x-coordinate; UTM easting in meters. Same as for 2nd return x-coordinate, the UTM zone code, 17 for this example, must be removed from the end of the data values.

- Column 7 (C7): 1st return y-coordinate; UTM northing in meters.

- Column 8 (C8): 1st return z-value; elevation in meters

- Column 9 (C9): 1st return intensity

*Note:
This ALSM system records two returns, 1st return and 2nd return also called first stop and last stop. For example, when a pulse is output by the laser, part of that pulse can reflect off an object closer to the aircraft such as a leaf on top of a tree while a portion of the pulse continues through the canopy and might for instance reflect off the hard ground. Since the pulse that reflected off the leaf on the top of the tree is closer to the aircraft, its return will reach the receiver first and be recorded first and is therefore labeled as the 1st return. The other portion of the pulse that continued past the tree canopy and reflected off the ground reaches the receiver after the first return and is therefore labeled as the 2nd return. If only one return is recorded by the receiver per

a pulse, that return is stored solely as a 2nd return value, no 1st return value is recorded. This was the case for the data record for row 2 above.

Appendix B

MatLab code for the function “GridData1”

```
function [XI,YI,ZI] = GridData1(file_name,grid_size,intmode,zone_code)
%=====================================================================================
==========
%Function:  GridData1
%Purpose:   Grids 9-column LiDAR data to user specifed grid size and interpolation
method where
%           9-column format = [time, 2nd-return x, y, z, intensity, 1st-return, x, y,
z, intensity];
%           User can grid 1st or 2nd return height or intensity. Program prompts user
to select choice.
%Inputs:    data file name including extension;
%           grid size (m);
%           interpolation mode -->'linear' standard linear interpolation
%            method,'invdist' - inverse distance,'cubic'
%           - cubic interpolation, 'nearest' - nearest neighbor;
%           UTM zone code x 10^6 to subtract from easting values, set to 0
%           if already subtracted from easting values.
%Outputs:   the function returns a 3-D array for easting x-values in meters,y-values
in meters, and the z-interpolated height or
%           intensity values [X,Y,Z]; A 3-D surface plot is also output.
%
%Example Function Call:    [X,Y,Z] = GridData1('DataFile.all',3,'linear',17000000);
%
%Created by: Michael Starek
%NCALM Center - University of Florida, 2005
%for help -->email "mstarek@ufl.edu"
%tested on Windows XP Machine with 1GB RAM running MatLab V6 Release 13, Service Pack1
%Known Problems: Large data files exceed MatLab's max variable size for InvDist
interpolation function;
%MatLab's Nearest neighbor function can sometimes produce erratic results dependenent
on nature of data set.
%=====================================================================================
===========


warning off MATLAB:griddata:DuplicateDataPoints;
%read in LIDAR data from file
InData = dlmread(file_name);
%change data points with no first pulse to last pulse xyz-intensity
InData(InData(:,6) == 0,6) = InData(InData(:,6) == 0,2);
InData(InData(:,7) == 0,7) = InData(InData(:,7) == 0,3);
InData(InData(:,8) == 0,8) = InData(InData(:,8) == 0,4);
InData(InData(:,9) == 0,9) = InData(InData(:,9) == 0,5);

'file loaded'

%load data and store in separate variables

x2 = [InData(:,2)];  %last stop
```

```
y2 = [InData(:,3)];
z2 = [InData(:,4)];
i2 = [InData(:,5)];
x1 = [InData(:,6)];  %first stop
y1 = [InData(:,7)];
z1 = [InData(:,8)];
i1 = [InData(:,9)];

%subtract out UTM zone-code
if (zone_code)
    x2=x2-zone_code; %gets rid of UTM coordinate zone code
    x1=x1-zone_code;
end

%scale range of easting and northing to start at 0 - reduces delaunay triangulation
errors for function used by MatLab;
minx1 = min(x1);
minx2 = min(x2);
miny1 = min(y1);
miny2 = min(y2);
x1= x1-minx1;
x2= x2-minx2;
y1= y1-miny1;
y2= y2-miny2;


%find max and mins of each data_set
XLimMin1=floor(min(x1)); XLimMax1=round(max(x1)); YLimMin1=floor(min(y1));
YLimMax1=round(max(y1));
XLimMin2=floor(min(x2)); XLimMax2=round(max(x2)); YLimMin2=floor(min(y2));
YLimMax2=round(max(y2));

choice = input('Please enter "1" to grid 1st-return or "2" to grid 2nd-return values:
');

if choice == 1
    decision = input('Please enter "1" to grid height values or "2" to grid
intensities: ');

    if decision == 1

%======================================================================================
    %GRIDDED SURFACE PLOT OF 1st-Stop HEIGHTS

%======================================================================================
    [XI,YI] = meshgrid(XLimMin1:grid_size:XLimMax1,YLimMin1:grid_size:YLimMax1);
    [XI,YI,ZI]  = griddata(x1,y1,z1,XI,YI,intmode);
    XI = XI + floor(minx1);
    YI = YI + floor(miny1);
    figure(1);
    surf(XI,YI,ZI); hold off; grid off;
           title('Surface Plot of Gridded 1st-Return Height Values'),shading interp,
view(-17,60), colormap(cool),
           xlabel('X-axis (meters)'),ylabel('Y-axis (meters)'),zlabel('Height
(meters)'),
           axis tight;

%======================================================================================
    elseif decision == 2

%======================================================================================
    %GRIDDED SURFACE PLOT OF 1st-Stop Intensities
```

```
%====================================================================================
    [XI,YI] = meshgrid(XLimMin1:grid_size:XLimMax1,YLimMin1:grid_size:YLimMax1);
    [XI,YI,ZI]  = griddata(x1,y1,i1,XI,YI,intmode);
    XI = XI + floor(minx1);
    YI = YI + floor(miny1);
    figure(1);
    surf(XI,YI,ZI); hold off; grid off;
           title('Surface Plot of Gridded 1st-Return Intensity Values'),shading
interp, view(-17,60), colormap(cool),
           xlabel('X-axis (meters)'),ylabel('Y-axis (meters)'),zlabel('Intensity'),
           axis tight;

%====================================================================================
    else
    'Input Error - Please reload the function and try again!'
    end

elseif choice == 2
    decision = input('Please enter "1" to grid height values or "2" to grid
intensities: ');
    if decision == 1

%====================================================================================
    %GRIDDED SURFACE PLOT OF 2nd-Stop HEIGHTS

%====================================================================================
    [XI,YI] = meshgrid(XLimMin2:grid_size:XLimMax2,YLimMin2:grid_size:YLimMax2);
    [XI,YI,ZI]  = griddata(x2,y2,z2,XI,YI,intmode);
    XI = XI + floor(minx2);
    YI = YI + floor(miny2);
    figure(1);
    surf(XI,YI,ZI); hold off; grid off;
           title('Surface Plot of Gridded 2nd-Return Height Values'),shading interp,
view(-17,60), colormap(cool),
           xlabel('X-axis (meters)'),ylabel('Y-axis (meters)'),zlabel('Height
(meters)'),
           axis tight;

%====================================================================================
    elseif decision == 2

%====================================================================================
    %GRIDDED SURFACE PLOT OF 2nd-Stop Intensities

%====================================================================================
    [XI,YI] = meshgrid(XLimMin2:grid_size:XLimMax2,YLimMin2:grid_size:YLimMax2);
    [XI,YI,ZI]  = griddata(x2,y2,i2,XI,YI,intmode);
    XI = XI + floor(minx2);
    YI = YI + floor(miny2);
    figure(1);
    surf(XI,YI,ZI); hold off; grid off;
           title('Surface Plot of Gridded 2nd-Return Intensity Values'),shading
interp, view(-17,60), colormap(cool),
           xlabel('X-axis (meters)'),ylabel('Y-axis (meters)'),zlabel('Intensity'),
           axis tight;

%====================================================================================

    else
    'Input Error - Please reload the function and try again!'
    end
```

```
else
    'Input Error - Please reload the function and try again!'
end
```